

REMARKS

This paper is submitted in reply to the Office Action dated December 28, 2007, within the three-month period for response. Reconsideration and allowance of all pending claims are respectfully requested.

In the subject Office Action, claims 1-5, 8-15 and 18-23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,131,112 to Bartz et al. and U.S. Patent Application Publication No. 2003/0167446 by Thomas, and further in view of “Version Management with CVS” by Per Cederqvist (Cederqvist). Claims 6-7 and 16-17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bartz et al., Thomas and Cederqvist, and further in view of U.S. Patent No. 6,385,768 to Ziebell.

Applicant respectfully traverses the Examiner's rejections to the extent that they are maintained. Applicant has amended claim 1, canceled claims 3-4 and 11-21, and added new claims 24-26. Applicant respectfully submits that no new matter is being added by the above amendments, as the amendments are fully supported in the specification, drawings and claims as originally filed. Applicant also notes that the amendments made herein are being made only for facilitating expeditious prosecution of the aforementioned claimed subject matter. Applicant is not conceding in this application that the originally claimed subject matter is not patentable over the art cited by the Examiner, and Applicant respectfully reserves the right to pursue this and other subject matter in one or more continuation and/or divisional patent applications.

Now turning to the rejections, and specifically to the rejection of independent claim 1, the Examiner has now added a third reference, Cederqvist, to the prior rejection based on Bartz and Thomas. The Examiner asserts that Cederqvist discloses the concept of adapting a standard code base to operate on a particular type of computer at page 2, paragraph 1, (“If you modify a program to better fit your site, you probably want to include your modifications when the next release of the program arrives.”)

It is important to note, however, that while Cederqvist arguably discloses incorporating third party revisions into a branch of a string of revisions in a code base,

Cederqvist does not disclose adapting a “standard” code base to work with a particular “type” of computer, as is presently recited in claim 1. The disclosure cited by the Examiner relates only to modifying a program to “better fit your site,” which Applicant submits constitutes at most a suggestion that programs can be customized for particular end user applications. There is nothing in this passage that suggests that a standard code base, which when read in the context of Applicant’s specification, is the type of code base that is capable of running on different types of computers, can be adapted to run on a particular “type” of computer. The more logical interpretation of Cederqvist is that the program that is being modified is already capable of running on a computer at the site, and the modifications merely customize that program to “better fit” the site, but do not specifically “adapt” that program to run on the computer at the site.

A “type” of computer, when read in the context of Applicant’s specification, is a particular computer platform, e.g., a particular processor type, a particular operating system, etc. Thus, put another way, Cederqvist does not disclose or suggest any “adaptation” of a program to operate on a particular computer platform, or “type” of computer. The changes are merely customizations that perhaps change the functionality of the program, but they do not appear to be the type of changes that are necessary to adapt a program to run on a specific platform. Cederqvist thus does not remedy the shortcomings of Bartz and Thomas, and claim 1 remains non-obvious over the prior art of record.

The Examiner will also note that Applicant has added the subject matter of claim 4 to claim 1, and as such has canceled claims 3-4. Therefore, claim 1 as amended now additionally recites:

- canonically parsing an intermediate version of the first release of the standard code base to generate a canonically-parsed representation of the intermediate version;
- generating the difference data includes comparing the canonically-parsed representations of the intermediate and modified versions of the first release of the standard code base;
- the intermediate version of the first release of the standard code base is generated using automated source transformation; and

- the modified version of the first release of the standard code base is generated by applying manual changes to the intermediate version of the first release of the standard code base.

In rejecting claims 3 and 4, the Examiner relies on Figs. 3-4 and 8 and col. 5, lines 10-35, col. 6, line 30 to col. 7, line 6, col. 8, lines 64-65 and col. 9, lines 46-66 of Bartz. Applicant respectfully submits, however, that these passages, as well as Bartz as a whole, do not disclose or suggest the concept of generating the modified version based upon applying manual changes to an intermediate version that has been generated using automated source transformation, as well as the concept of generating difference data by comparing canonically-parsed representations of the intermediate and modified versions.

While Bartz may arguably disclose code that is analogous to an “intermediate” representation, Applicant notes that claim 1 does not merely recite the use of intermediate representations. Instead, claim 1 focuses on the generation of difference data between (1) an intermediate version of a standard code based that has been generated using automated source transformation, and (2) a modified version that has been generated by applying manual changes to the intermediate version. As shown in Fig. 2 of the published application, for example, an intermediate version L_N may be generated from a standard version S_N using automated source transformation, and then a modified version M_N may be generated from the intermediate version L_N via manual changes, and difference data is generated between I_N and M_N (see also Fig. 3). Note, however, that the modified version M_N is not directly generated from the standard version S_N, and furthermore, the difference data is not between I_N and S_N, or between M_N and S_N.

Bartz, in contrast, discloses in the cited passages associated with Figs. 3 and 4 the comparison of different versions created by different developers (see, e.g., col. 6, lines 30-50). Both versions therefore likely represent manual changes made off of the same root version. Likewise, the cited passages associated with Figs. 7-8 are associated with a concurrent building and development process where reference builds of a project are created on a periodic basis. Individual developers are able to retrieve copies of files into local enlistment areas and make changes to those files. When the files are later retrieved

from a common area, the builder's local changes are applied to those files for use by the developer.

There is no disclosure, however, in any of these passages of any functionality for generating an intermediate version from a release of a code base using automated source transformation, further modifying the intermediate version via manual changes to create a modified version, and then generating difference data between those intermediate and modified versions. Accordingly, Bartz falls short of disclosing or suggesting this combination of features.

Neither Thomas nor Cederqvist is relied upon by the Examiner for disclosing or suggesting the generation of difference data between an intermediate version of a standard code based that has been generated using automated source transformation, and a modified version that has been generated by applying manual changes to the intermediate version. In addition, Applicant can find no disclosure or suggestion in either reference related to this concept. Accordingly, Applicant submits that the proposed combination falls short of disclosing or suggesting this concept, and thus falls short of disclosing or suggesting each and every feature of claim 1.

Applicant therefore respectfully submits that claim 1 is non-obvious over Bartz, Thomas, Cederqvist and the other prior art of record. Reconsideration and allowance of claim 1, and of claims 2, 5-10 and 23-26 which depend therefrom, are therefore respectfully requested.

As a final matter, the Examiner will note that Applicant has added new claims 24-26. Claim 24 depends from claim and additionally recites that the standard code base comprises system-independent program code configured to execute on a plurality of types of computers, and that the modified version is adapted from the first release of the standard code base for the purpose of adapting the first release of the standard code base to operate on a particular type of computer. Support for this claim may be found, for example, in paragraphs [0006]-[0008] of the published Application. The Examiner has admitted that Bartz and Thomas do not disclose or suggest "standard" code bases or adaptations of the same for a particular type of computer, and has relied on Cederqvist for allegedly

disclosing this feature. However, there is nothing in this reference that discloses or suggests a standard code base of “system-independent” program code configured to execute on a plurality of types of computers, and an adaptation of the same to operate on a particular “type” of computer. Claim 24 is therefore additionally patentable over the prior art of record for this reason.

Claim 25 depends from claim 1 and additionally recites that the standard code base comprises a library configured to run on a single-user computer, and that the modified version is adapted from the first release of the standard code base for the purpose of adapting the first release of the standard code base to operate in a client/server environment. Support for this claim may be found, for example, in paragraphs [0008] and [0031] of the published Application. None of the references cited by the Examiner discloses or suggests the concept of adapting a library that is configured to run on a single-user computer to operate in a client/server environment. Claim 25 is therefore additionally patentable over the prior art of record for this reason.

Claim 26 depends from claim 25 and additionally recites that the standard code base comprises a graphical user interface library configured to display graphical user interface components on a same computer upon which the standard code base is run, and that the modified version is adapted from the first release of the standard code base for the purpose of adapting the graphical user interface library in the first release of the standard code base to initiate a display of graphical user interface components on a different computer from which the standard code base is run. Support for this claim may be found, for example, in paragraphs [0008] and [0031] of the published Application. None of the references cited by the Examiner discloses or suggests the concept of adapting a GUI library that is configured to display components on a local computer to instead initiate the display of components on a different computer (e.g., a remote computer in a client/server environment). Claim 26 is therefore additionally patentable over the prior art of record for this reason.

In summary, Applicant respectfully submits that all pending claims are novel and non-obvious over the prior art of record. Reconsideration and allowance of all pending

claims are therefore respectfully requested. If the Examiner has any questions regarding the foregoing, or which might otherwise further this case onto allowance, the Examiner may contact the undersigned at (513) 241-2324. Moreover, if any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23 3000.

Respectfully submitted,

March 28, 2008
Date

/Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234